WE CLAIM

1. Apparatus for processing data under control of a set of program instructions that map upon interpretation to data processing operations to be performed, said apparatus comprising:

(i) a fixed mapping hardware interpreter operable to interpret a fixed mapping group of said set of program instructions, whereby a program instruction from said fixed mapping group maps to a fixed sequence of one or more data processing operations; and

(ii) a programmable mapping hardware interpreter operable to interpret a programmable mapping group of said program instructions, whereby a program instruction from said programmable mapping group maps to a sequence of one or more data processing operation that varies in dependence upon programming of said programmable mapping hardware interpreter.

2. Apparatus as claimed in claim 1, further comprising a software execution unit operable to interpret at least a software interpreted group of program instructions.

3. Apparatus as claimed in claim 2, wherein said software execution unit is one, or a combination of:

(i) a software interpreter; and

(ii) a just in time compiler.

4. Apparatus as claimed in claim 1, wherein said programmable mapping hardware interpreter has a fixed set of sequences of one or more data processing operations to which program instructions from said programmable mapping group may be mapped.

5. Apparatus as claimed in claim 1, wherein said programmable mapping hardware interpreter includes a programmable translation table that translates program instructions within said programmable mapping group into a sequence of one or more data processing operation to be performed.

6. Apparatus as claimed in claim 5, wherein said sequence of one or more data processing operations to be performed is specified with an operation value within said programmable translation table.

7. Apparatus as claimed in claim 6, wherein said programmable translation table is a content addressable memory addressed via a program instruction value to specify a corresponding operation value.

8. Apparatus as claimed in claim 6, wherein said programmable translation table is a random access memory with a program instruction value being decoded to address a storage location within said random access memory for a corresponding operation value.

9. Apparatus as claimed in claim 5, wherein said programmable translation table includes an invalid entry trap operable to block storage of unsupported mappings within said translation table.

10. Apparatus as claimed in claim 1, wherein said sequences of one or more data processing operations each comprise processing operations equivalent to one or more native program instructions of a processor core that is a target for said fixed mapping hardware interpreter and said programmable mapping hardware interpreter.

11. Apparatus as claimed in claim 1, wherein said program instructions are Java bytecodes.

12. Apparatus as claimed in claim 1, wherein said software interpreted group includes all those instructions not within said fixed mapping group or said programmable mapping group.

13. Apparatus as claimed in claim 12, wherein said software interpreted group includes all of said program instructions, said software interpreter being invoked when neither said fixed mapping hardware interpreter or said programmable mapping hardware interpreter can interpret a program instruction.

14. Apparatus as claimed in claim 1, wherein said fixed mapping hardware interpreter and said programmable mapping hardware interpreter share at least some decoder hardware.

15. Apparatus as claimed in claim 1, comprising a translation pipeline stage with a program instruction buffer operable to store program instructions to be interpreted providing an input to said translation pipeline stage such that program instructions are subject to a programmable mapping within said translation pipeline stage prior to further interpretation.

16. A method of processing data under control of a set of program instructions that map upon interpretation to data processing operations to be performed, said method comprising the steps of:

(i) using a fixed mapping hardware interpreter to interpret a fixed mapping group of said set of program instructions, whereby a program instruction from said fixed mapping group maps to a fixed sequence of one or more data processing operations; and

(ii) using a programmable mapping hardware interpreter to interpret a programmable mapping group of said program instructions, whereby a program instruction from said programmable mapping group maps to a sequence of one or more data processing operations that varies in dependence upon programming of said programmable mapping hardware interpreter.

17. A method as claimed in claim 16, further comprising using a software execution unit to interpret at least a software interpreted group of program instructions.

18. A method as claimed in claim 17, wherein said software execution unit is one, or a combination of:

(i) a software interpreter; and

(ii) a just in time compiler.

19. A method as claimed in claim 1, wherein said programmable mapping hardware interpreter has a fixed set of sequences of one or more data processing

operations to which program instructions from said programmable mapping group may be mapped.

20. A method as claimed in claim 1, wherein said programmable mapping hardware interpreter includes a programmable translation table that translates program instructions within said programmable mapping group into a sequence of one or more data processing operations to be performed.

21. A method as claimed in claim 20, wherein said sequence of one or more data processing operations to be performed is specified with an operation value within said programmable translation table.

22. A method as claimed in claim 21, wherein said programmable translation table is a content addressable memory addressed via a program instruction value to specify a corresponding operation value.

23. A method as claimed in claim 21, wherein said programmable translation table is a random access memory with a program instruction value being decoded to address a storage location within said random access memory for a corresponding operation value.

24. A method as claimed in claim 20, wherein said programmable translation table includes an invalid entry trap operable to block storage of unsupported mappings within said translation table.

25. A method as claimed in claim 16, wherein said sequences of one or more data processing operations each comprise processing operations equivalent to one or more native program instructions of a processor core that is a target for said fixed mapping hardware interpreter and said programmable mapping hardware interpreter.

26. A method as claimed in claim 16, wherein said program instructions are Java bytecodes.

27. A method as claimed in claims 16, wherein said software interpreted group includes all those instructions not within said fixed mapping group or said programmable mapping group.

28. A method as claimed in claim 27, wherein said software interpreted group includes all of said program instructions, said software interpreter being invoked when neither said fixed mapping hardware interpreter or said programmable mapping hardware interpreter can interpret a program instruction.

29. A method as claimed in claim 16, wherein said fixed mapping hardware interpreter and said programmable mapping hardware interpreter share at least some decoder hardware.

30. A method as claimed in claim 16, comprising a translation pipeline stage with a program instruction buffer operable to store program instructions to be interpreted providing an input to said translation pipeline stage such that program instructions are subject to a programmable mapping within said translation pipeline stage prior to further interpretation.

31. A computer program product for controlling a data processing apparatus to provide interpretation of a set of program instructions that map upon interpretation to sequences of one or more data processing operations to be performed, said computer program product comprising:

(i) mapping configuration logic operable to program a programmable mapping hardware interpreter to interpret a programmable mapping group of program instructions, whereby a program instruction from said programmable mapping group maps to a sequence of one or more data processing operation that varies in dependence upon programming of said programmable mapping hardware interpreter.

32. A computer program product as claimed in claim 31, wherein said programmable mapping hardware interpreter has a fixed set of data processing operations to which program instructions from said programmable mapping group may be mapped.

33. A computer program product as claimed in claim 31, wherein said programmable mapping hardware interpreter includes a programmable translation table that translates program instructions within said programmable mapping group into a sequence of one or more data processing operation to be performed.

34. A computer program product as claimed in claim 33, wherein said sequence of one or more data processing operations to be performed is specified with an operation value within said programmable translation table.

35. A computer program product as claimed in claim 33, wherein said programmable translation table includes an invalid entry trap operable to block storage of unsupported mappings within said translation table.

36. A computer program product as claimed in claim 31, wherein said sequence of one or more data processing operations each comprise processing operations equivalent to one or more native program instructions of a processor core that is a target for said programmable mapping hardware interpreter.

37. A computer program product as claimed in claim 31, wherein said program instructions are Java bytecodes.

38. A computer program product as claimed in claim 31, comprising a software execution unit operable to interpret program instructions.

39. A computer program product as claimed in claim 38, wherein said software execution unit is one, or a combination, of:

(i) a software interpreter; and

(ii) a just in time compiler.